Capítulo 1

LaTeX 2ε

¿Qué necesita para ser funcionar LaTeX 2ε?
—What does LaTeX 2ε need to work?

Este capítulo establece los recursos necesarios para el desarrollo de documentos con el sistema LaTeX 2ε. Esta trata los Fundamentos, Materiales e instalación, Montaje y Procesamiento del primer documento, Procedimiento sintaxis del lenguaje; todo lo necesario para arrancar desde un punto cero, (i.e., una introducción completa) técnicamente hablando.

1.1. Fundamentos

1.1.1. Versión

TeX (Typesetting system) 1977 (inicio), 1982 (lanzamiento), 1989 (última mejora).

Desarrollado por Donald Ervin Knuth en 1982 como "un nuevo sistema de composición tipográfica [—typesetting system] para la creación de libros hermosos —y especialmente para libros que contienen mucha matemática." [1].

Actualmente es una marca registrada de la American Mathematical Society (AMS).

LaTeX 2.09 (Document Preparation System) 1985-09-01.

Desarrollado por Leslie Lamport en 1985 [2] (inicialmente llamado *Document Preparation System*) para ser un lenguaje de alto nivel, que internamente reduce el código al lenguaje de bajo nivel TeX.

LaTeX 2ε (new Document Preparation System) 1994-06-01 to 2020-05-25.

Desarrollado por el *LaTeX Project Team* en 1994 [3] para ser la nueva versión del estándar LaTeX (Document Preparation System).

LaTeX3 (The long-term future of LaTeX). 2009 (inicio), 2017-12-16 to 2021-02-18.

Desarrollado por el *LaTeX3 Project Team* desde el 2009 [4] para ser el nuevo lenguaje de programación de paquetes y clases para LaTeX.

1.1.2. Cualidades

- Software libre
- Independiente de la plataforma, la salida es la misma en todas las plataformas
- Permite el control más fino sobre cualquier aspecto tipográfico del documento
- No es un procesador de texto WYSIWYG (“what you see is what you get”), sino un lenguaje de programación interpretado.
- Diseñado para escribir el documento según el diseño lógico YAFIYGI (“you asked for it; you got it”) a diferencia de WYSIWYG de diseño visual.

1.1.3. Plataforma

La plataforma de desarrollo basicamente esta compuesto por:

- Motor —Composition engine
- Lenguaje —Language programming layer (loader)
- Bibliotecas —Packages

Una distribución de TeX trae consigo el motor y los paquetes básicos. Distribuciones de TeX:

MiKTeX miktex.org/download
(Windows) A separate distribution entirely that

TeX Live www.tug.org/texlive/
(Windows, Linux, and OS X), the standard, cross-platform distribution.

MacTeX www.tug.org/mactex/
(Mac OS X) A packaged version of TeX Live made for OS X with some Mac-specific tools

1.2. Entorno de desarrollo

1.2.1. Instalación del sistema

La Figura 1.1 muestra la documentación de la distribución MiKTeX (miktex.chm→P.689) [5] y TeX Live (readme.es.html→P.689).

1.2.2. Programas de administración

Programas de administración y configuración del sistema.

Administrador – TeX Live

```
>_tlmgr [option...] action [option...] [operand...]
```

TeX Live Manager. Este programa de consola es el administrador de opciones de TeX Live.

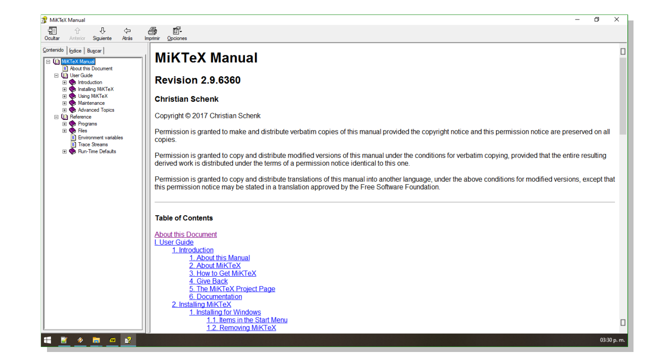
Alguna de las opciones son:

- `--version`
- `--help`

Figura 1.1. Documentación de la distribución instalada

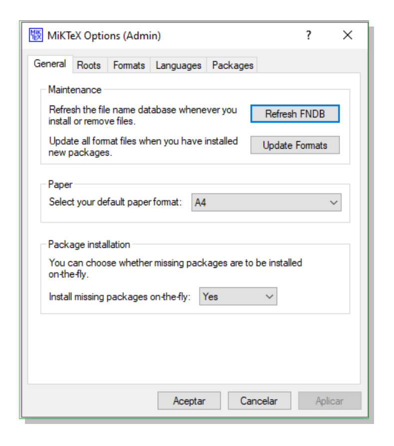


a). Linux – TeX Live 2018 instalado



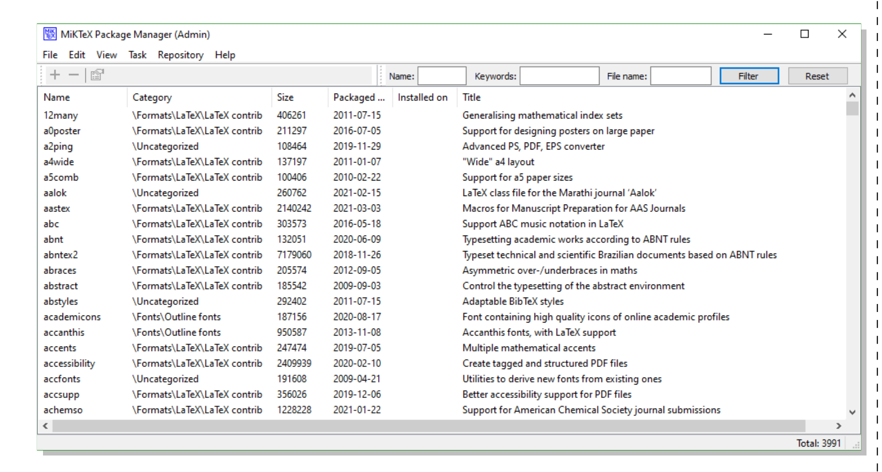
b). Windows – MiKTeX 2.9 instalado

Figura 1.2. MiKTeX – Maintenance





a). MiKTeX Options (Admin)



Name	Category	Size	Packaged ...	Installed on	Title
12many	\Formats\LaTeX\LaTeX contrib	406261	2011-07-15		Generalising mathematical index sets
a0poster	\Formats\LaTeX\LaTeX contrib	211297	2016-07-05		Support for designing posters on large paper
a2ping	\Uncategorized	108464	2019-11-29		Advanced PS, PDF, EPS converter
a4wide	\Formats\LaTeX\LaTeX contrib	137197	2011-01-07		"Wide" a4 layout
a5comb	\Formats\LaTeX\LaTeX contrib	100406	2010-02-22		Support for a5 paper sizes
aalok	\Uncategorized	260762	2021-02-15		LaTeX class file for the Marathi journal 'Aalok'
aastex	\Formats\LaTeX\LaTeX contrib	2140242	2021-03-03		Macros for Manuscript Preparation for AAS Journals
abc	\Formats\LaTeX\LaTeX contrib	303573	2016-05-18		Support ABC music notation in LaTeX
abnt	\Formats\LaTeX\LaTeX contrib	132051	2020-06-09		Typesetting academic works according to ABNT rules
abntex2	\Formats\LaTeX\LaTeX contrib	7179060	2018-11-26		Typeset technical and scientific Brazilian documents based on ABNT rules
abraces	\Formats\LaTeX\LaTeX contrib	205574	2012-09-05		Asymmetric over-/underbraces in maths
abstract	\Formats\LaTeX\LaTeX contrib	185542	2009-09-03		Control the typesetting of the abstract environment
abstyles	\Uncategorized	292402	2011-07-15		Adaptable BibTeX styles
academicons	\Fonts\Outline fonts	187156	2020-08-17		Font containing high quality icons of online academic profiles
accanthis	\Fonts\Outline fonts	950587	2013-11-08		Accanthis fonts, with LaTeX support
accents	\Formats\LaTeX\LaTeX contrib	247474	2019-07-05		Multiple mathematical accents
accessibility	\Formats\LaTeX\LaTeX contrib	2409939	2020-02-10		Create tagged and structured PDF files
accfonts	\Uncategorized	191608	2009-04-21		Utilities to derive new fonts from existing ones
accsupp	\Formats\LaTeX\LaTeX contrib	356026	2019-12-06		Better accessibility support for PDF files
achemso	\Formats\LaTeX\LaTeX contrib	1228228	2021-01-22		Support for American Chemical Society journal submissions

b). MiKTeX Package Manager (Admin)

```
# tlmgr --version
(running on Debian, switching to user mode!)
tlmgr revision 47951 (2018-06-07 07:55:43 +0200)
tlmgr using installation: /usr/share/texlive
TeX Live (http://tug.org/texlive) version 2018
```

Administrador – MiKTeX

>_mo.exe

MiKTeX Options. Este programa ejecutable (mo.exe→P.689, mo_admin.exe) [5, p. 15] es el administrador de opciones generales de MiKTeX.

En la Figura 1.2 el b) es el administrador de paquetes que se puede invocar vía: `mpm_mfc_admin` Instalación de paquetes automática.

1.2.3. Instalación del IDE

Desarrollo integrado (IDE): Editores especializados para código LaTeX [6, p. 5] que ayudan en el desarrollo:

Capítulo 2

Interfaces fundamentales

Interfaces de composición tipográfica
—A TEX programming environment

Este capítulo establece los conocimientos fundamentales para empezar a escribir código en los lenguajes con base TEX, iniciando en como se interpreta un carácter del código fuente, su composición como símbolo renderizado y posterior ubicación en el texto.

2.1. El Manejo del texto

2.1.1. Terminología

Los términos *carácter*, *glifo* y *punto de código* (char number) se usan según lo definido por Unicode [10, §1.3].

2.1.2. Carácter

Un *carácter* es una unidad atómica de texto.[11, §11.1.1]

`\char`⟨*number*⟩

Comando primitivo [1, p. 286, 43], expande el glifo del carácter ⟨*number*⟩ de la fuente actual.

⟨*number*⟩ es un número entero de 8-bit's (0 a 255), si excede esta capacidad se produce un error (`! Bad character code`) en tiempo de compilación.

Por ejemplo, el carácter '@' (U+0040, decimal 64):

@

```
\char64
```

Un *carácter de control* puede ser escrito con la secuencia:

`^^`⟨*tokens*⟩

Esta secuencia representa un carácter de control. [1, p. 368]

Donde ⟨*tokens*⟩ puede ser: *una letra mayúscula* (Ctrl-letter), del código de ASCII; o dos *dígitos hexadecimales*.

Por ejemplo, `^^M`→P.27 y `^^0d` es el carácter de control CR (U+000D).

13, 13	`\number`\^^M,` `\number`\^^0d`

`\number`⟨*number*⟩

Comando primitivo [1, p. 213, 472], expande la representación base decimal de ⟨*number*⟩→P.196.

64	`\number`@`

918000	`\number"E01F0`

2.1.3. Código de categoria

`\catcode`⟨*carácter*⟩=⟨*categoria*⟩

Comando primitivo TeX, establece la ⟨*categoria*⟩ del ⟨*carácter*⟩.

⟨*carácter*⟩: es un número entero de 8-bits, véase `\char`→P.19.

⟨*categoria*⟩: es un número entero entre 0 a 15, véase Tabla 2.1. Para obtener la ⟨*categoria*⟩ puede usar `\number`→P.20 o `\the`→P.228.

12	`\number\catcode64`

La clasificación del ⟨*carácter*⟩ es vigente con visibilidad precedente en el grupo local.

@ es de categoria 11

```
{
    \catcode64=11
    \char64\ es de categoria \the\catcode64
}
```

La Tabla 2.1 (latex.ltx→P.689, Ln: 282) [7, p. 14] [1, p. 37] muestra la clasificación de los caracteres, con el comando `\catcode` en 16 categorias numeradas del 0 al 15.

Tabla 2.1. Códigos de categoria

Código	Categoria	Carácter	TeX plain	Unicode
0	Carácter de escape	\		005C
1	Inicio de grupo	{		007B
2	Fin de grupo	}		007D
3	Cambio matemático	$		0024
4	Alineador de tabulación	&		0026
5	Fin de línea	CR	^^M	000D
6	Parámetro de macro	#		0023
7	Exponente	^		005E
8	Subíndice	_		005F
9	Carácter ignorado	NUL	^^@	0000

Capítulo 3

Composición matemática

Composición tipográfica para escribir matemáticas
—Typesetting standard math

3.1. Modo matemático

Modos para escribir material matemático.

3.1.1. Math mode

(Math mode in-line formulas). Modo lísta horizontal.

`$`⟨*formula*⟩`$` (`\catcode`→P.20: 3)

Token '`$`' (en la apertura) pasa al modo matematico o (en el cierre) retorna al modo previo.

Lorem ipsum $m=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$ dolor sit amet

```
Lorem ipsum $m=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$ dolor sit amet
```

`\(`⟨*formula*⟩`\)`

Produce igual a `$`⟨*formula*⟩`$` [7, p. 275].

Lorem ipsum $m=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$ dolor sit amet

```
Lorem ipsum \(m=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}\) dolor sit amet
```

`\begin{math}`
⟨*contenido del entorno*⟩
`\end{math}`

Alias de `\(...\)` [7, p. 277].

PRINT PREVIEW (BETA TEST) ~ PRESENTACIÓN PRELIMINAR

Lorem ipsum $m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$ dolor sit amet

```
Lorem ipsum \begin{math}m=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}\end{math} dolor sit amet
```

`\ensuremath`⟨*formula*⟩

En LR o modo párrafo, es equivalente a `$`⟨*formula*⟩`$` [7, p. 279] [2, p. 53, 187] en math mode equivalente a ⟨*formula*⟩.

Lorem x^2 ipsum	`Lorem \ensuremath{x^2} ipsum`

`\ifmode`→P.207 Para evaluar si esta en modo matemático.

3.1.2. Display math mode

(Display math mode). Modo presentación.

`$$`⟨*formula*⟩`$$`

Lorem ipsum dolor sit amet,

$$m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$$

consectetuer adipiscing elit.

```
Lorem ipsum dolor sit amet,
$$m=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$$
consectetuer adipiscing elit.
```

`\[`⟨*formula*⟩`\]`

Produce igual a `$$`⟨*formula*⟩`$$` [7, p. 276, 280].

$$m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$$

```
\[m=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}\]
```

`\begin{displaymath}`
⟨*contenido del entorno*⟩
`\end{displaymath}`

Disfraz de `\[`⟨*formula*⟩`\]` [7, p. 277][2, p. 188].

$$m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$$

```
\begin{displaymath}
    m=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
\end{displaymath}
```

`\lefteqn`⟨*formula*⟩

Macro [7, p. 279] expande a `\rlap{$\displaystyle #1$}`; ancla la ⟨*formula*⟩, en modo display, a la derecha del punto actual.

====.$\sqrt{\frac{1}{2}}$====

```
====.\lefteqn{\sqrt\frac12}====
```

$m = \frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}$

```
\lefteqn{m=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}}
```

3.1.3. Every math mode

`\everymath={`⟨*formulas*⟩`}` (initially `={}`)

Este primitivo, registrador (del tipo *toks*), inserta las ⟨*formulas*⟩ en el inicio de cada *math mode*.

Esta macro es redefinido en LaTeX 2_ε [7, p. 159] con su nuevo default '`={\check@mathfonts\the \everymath}`'.

Lorem $\alpha \to 4$ ipsum $\alpha \to 7$

```
{
    \everymath={\alpha\to}
    Lorem $4$ ipsum \(7\)
}
```

`\everydisplay={`⟨*formulas*⟩`}` (initially `={}`)

Este primitivo, registrador (del tipo *toks*), inserta las ⟨*formulas*⟩ en el inicio de cada *display math mode*.

Esta macro es redefinido en LaTeX 2_ε [7, p. 159] con su nuevo default '`={\check@mathfonts\the \everydisplay}`'.

Lorem

$$\sqrt{4} \to y$$

ipsum

$$\sqrt{4} \to x$$

```
{
    \everydisplay={\sqrt4\to}
    Lorem $$y$$ ipsum \[x\]
}
```

3.2. Formulas referenciadas

3.2.1. Ecuaciones con caption

`\begin{equation}`
⟨*contenido del entorno*⟩
`\end{equation}`

Este entorno [7, p. 276, 280], [2, p. 188] es igual que `{displaymath}` pero la ecuación es numerada.

$$\theta = \frac{s}{R} \qquad (3.1)$$

```
\begin{equation}
    \theta = \frac sR
\end{equation}
```

Extendido: para una ecuación muy larga (más de una línea) use `multline`→P.272, del paquete `amsmath`→P.270.

Sucesión de ecuaciones y con alineamiento.

`\begin{eqnarray}`
⟨*contenido del entorno*⟩

Capítulo 4

Composición gráfica

Composición de gráficos vectoriales
—Composition of vectorial graphics

4.1. Modo gráfico

\begin{picture}(⟨w,h⟩)(⟨x_o,y_o⟩)
⟨*contenido del entorno*⟩
\end{picture}

Reserva un rectángulo invisible con base ⟨w⟩ y altura ⟨h⟩.

Origen del sistema: esquina inferior isquierda.

⟨x_o,y_o⟩: Asigna coordenadas arbitrarias a la esquina inferior isquierda.

```
\begin{picture}(60,40)
    \put (0,0){\line(0,1){40}}
    \put (0,0){\line(1,0){60}}
    \put (0,0){\circle{4}}
\end{picture}
```

```
\begin{picture}(60,40)(-40,-20)
    \put (0,-20){\line(0,1){40}}
    \put (-40,0){\line(1,0){60}}
    \put (0,0){\circle{4}}
\end{picture}
```

4.2. Atributos de dibujo

4.2.1. Unidad de longitud

`\unitlength` (default `1 pt`)

Este comando [7, p. 334] es un registrador tipo *dimen*, que almacena la unidad de longitud (medida física) que representan los valores escalares de `picture`.

PRINT PREVIEW (BETA TEST) ~ PRESENTACIÓN PRELIMINAR

```
\unitlength=1cm
\begin{picture}(3,2)
    \put (0,0){\line(0,1){2}}
    \put (0,0){\line(1,0){3}}
    \put (0,0){\circle{.2}}
\end{picture}
```

Visibilidad:

28.45274pt
1.0pt

```
{
    \unitlength=1cm
    \the\unitlength\\
}
\the\unitlength
```

4.2.2. Grosor de línea

Estas declaraciones se pueden usar múltiples veces en una gráfica para cambiar el grosor de líneas particulares. `\thinlines` es el default.

`\linethickness{⟨grosor⟩}`

Establece la anchura de líneas verticales y horizontales. Este no tiene efecto para líneas inclinadas o círculos.

```
\begin{picture}(40,30)
    \linethickness{.4pt}
    \put (0,0){\line(0,1){30}}
    \linethickness{1.6pt}
    \put (0,0){\line(1,0){40}}
\end{picture}
```

`\thinlines`

Líneas delgadas.

```
\begin{picture}(90,30)
    \thinlines
    \put (0,0){\line(4,3){40}}
    \thicklines
    \put (50,0){\line(4,3){40}}
\end{picture}
```

`\thicklines`

Líneas gruesas.

Capítulo 5

Programación

Programación de alto y bajo nivel
—Programming in LaTeX and TeX

El capítulo trata la programación, es decir, la *creación* y gestión de nuevas funciones en el lenguaje (i.e., formato, según `\fmtname`→P.14) Plain TeX (versión 3.141592653) y LaTeX 2ε (versión 2017-04-15 o superior).

5.1. TeX

TeX es un lenguaje de macro, crear nuevas funciones significa crear macros (o, en terminología LaTeX, comandos).

5.1.1. Definiciones (Macros)

Definiendo nuevas macros [1, p. 198] [19, p. 109].

1. `\def`→P.187 definición local
2. `\edef`→P.190 definición local expandido
3. `\gdef`→P.190 definición global
4. `\xdef`→P.191 definición global expandido

`\def`⟨*control sequence*⟩⟨*parameters*⟩`{`⟨*replacement text*⟩`}`

Comando primitivo TeX, define localmente la macro ⟨*control sequence*⟩ y su firma de ⟨*parameters*⟩ y a su invocación es expandido por ⟨*replacement text*⟩.

⟨*control sequence*⟩ es el identificador de la macro, formado según la sintaxis estándar (\⟨*csname*⟩→P.21); o puede ser un *token* de categoria 13.

⟨*parameters*⟩ es una expresión de tokens *patrón de coincidencia* que sirve para capturar los argumentos que recibe la macro. Los parámetros son `#1`, `#2` y asi susesivamente hasta un máximo de nueve. Si en la invocación los argumentos no cumplen el *patrón de coincidencia* se produce un error en tiempo de compilación.

⟨*replacement text*⟩ es una serie de tokens, la especificación de expansión, cuando la macro es invocada se reemplaza los ⟨*parameters*⟩ `#1`, `#2`, etc. por sus valores.

Executing it: 'Lorem ipsum'.

```
\def\macro{Lorem ipsum}
Executing it: `\macro'.
```

Buenas practicas de programación: con el fin de hacer mas legible el código puede usar saltos de línea (CR, U+000D y LF, U+000A) o identación (cuatro caracteres SP, U+0020), pero,

Executing it: ' Lorem ipsum '.

```
\def\macro{
    Lorem ipsum
}
Executing it: `\macro'.
```

debe tomar en cuenta como se interpretan ellos (véase `\space`→P.27), la secuencia de definición interpretado es:

`\def\macro{` SP `Lorem` SP `ipsum` SP `}`

El identificador ⟨*control sequence*⟩ puede ser dado como un simple token con categoria 13 (véase `\active`→P.30)

Executing it: 'LoXXm iXsXm'.

```
\catcode`*=13
\def*{X}
Executing it: `Lo**m i*s*m'.
```

Macro ⟨*parameters*⟩ token:

#⟨*número*⟩ (Token de *categoria*→P.20: 6)

El token # seguido de un ⟨*número*⟩ entre 1–9 se interpreta como parámetro de macro.

here is 'A'BC
here is 'AB'C

```
\def\macro#1{here is `#1'}
\macro ABC\\
\macro {AB}C
```

Interpretación de espacios entre ⟨*control sequence*⟩ y ⟨*parameters*⟩: los espacios despues de ⟨*control sequence*⟩ se depuran, hasta encontrar un token no espacio, desde ahi los espacios son parte del patrón de coincidencia para definir los ⟨*parámetros*⟩

here is "Lorem
"ipsum

```
\def\macro    #1   {here is ``#1''}
\macro Lorem\\
```

```
ipsum
```

la secuencia interpretada en la firma y en la invocación es:

\def\macro#1 SP {...}
\macro SP Lorem\\ SP ipsum.

Interpretación de otros caracteres en los ⟨*parámetros*⟩

here is “abc” and “def ghi” jkl mno.

```
\def\macro#1 #2,{here is ``#1'' and ``#2''}
\macro abc def ghi, jkl mno.

% \macro abc def ghi jkl mno.       %%No compila, error de firma
```

note que la coma se captura como token (es parte de la firma de `\macro`) y por tanto no existe en la impresión. La última línea no cumple la firma de la macro, y produce un error, en el archivo log y la terminal:

```
Runaway argument?
def ghi jkl mno.
! Paragraph ended before \macro was complete.
```

first ‘A’, second ‘BC’, third ‘DEF’.G

```
\def\macro#1#2..#3 {first `#1', second `#2', third `#3'.}
\macro ABC..DEF G
```

I found arguments ‘42’, ‘43’ and ‘44’.

```
\def\macro(#1,#2,#3){I found arguments `#1', `#2' and `#3'.}
\macro(42,43,44)
```

Macros definidos en múltiples niveles:

Jane dice: hola Eric.
Anita dice: adios Marcos.

```
\def\defmacro#1#2{%
   \def#1##1##2{##1 dice: #2 ##2.}%
}
\defmacro \hello   {hola}   % expands to: \def\hello#1#2{#1 dice: hola #2.}
\defmacro \goodbye {adios}
\hello    {Jane}   {Eric}\\
\goodbye  {Anita}  {Marcos}
```

Visibilidad: La definición de ⟨*control sequence*⟩ es visible en el grupo local

Capítulo 6

Lenguaje de programación LaTeX3

6.1. Introducción

6.1.1. Especificación

La especificación del lenguaje LaTeX3 se desarrolla con base la implementación (expl3.ltx→P.692) y la documentación oficial (source3.tex→P.692) [25].
Fundamentos de la especificación, por ejemplo para el comando `\ExplSyntaxOn`→P.262:

1. Readme

   se da una noción en [4, p. 13], [26, p. 624] y [27, p. 14]; y en la documentación oficial se presenta en (source3body.tex→P.692, ln: 271) [25, p. 3].

2. Source code

   Su código fuente reside en (expl3-code.tex→P.692, ln: 275);

3. Implementation

   Describe el código fuente en (l3bootstrap.dtx→P.692, ln: 620) [25, p. 269];

4. Documentation

   Documenta su sintaxis en (l3bootstrap.dtx→P.692, ln: 158) [25, p. 6].

6.1.2. Paquete `expl3` (LaTeX3 programming layer)

`\usepackage{expl3}`

Paquete versión 2021-02-18 (expl3.pdf→P.692, .sty) [27], descrito como *L3 programming layer (loader)*, todavía es experimental, pero se considera bastante estable.

1. `\ProvidesExplPackage`→P.261
2. `\ProvidesExplClass`→P.262
3. `\ProvidesExplFile`→P.262

`\ProvidesExplPackage{`⟨*nombre*⟩`}{`⟨*fecha*⟩`}{`⟨*versión*⟩`}{`⟨*descripción*⟩`}`

Este comando (expl3.sty→P.692, ln: 29) [25, p. 6] es similar al correspondiente `\ProvidesPackage`→P.245.
Version identifiers, Issue 11 [4, p. 23]. Por ejemplo,

Print Preview (Beta Test) ~ Presentación Preliminar

```
\ProvidesExplPackage{xparse}
                    {2021-03-12}{}
                    {L3 Experimental document command parser}
```

`\ProvidesExplClass{⟨nombre⟩}{⟨fecha⟩}{⟨versión⟩}{⟨descripción⟩}`

Este comando (expl3.sty→P.692, ln: 31) [25, p. 6] es similar al correspondiente `\ProvidesClass`→P.247.

`\ProvidesExplFile{⟨nombre⟩}{⟨fecha⟩}{⟨versión⟩}{⟨descripción⟩}`

Este comando (expl3.sty→P.692, ln: 33) [25, p. 6] es similar al correspondiente `\ProvidesFile`→P.245.

6.1.3. Entorno de programación LaTeX3

`\ExplSyntaxOn⟨código⟩\ExplSyntaxOff`

Estos comandos (expl3-code.tex→P.692, ln: 275) [25, p. 269, 6, 3] modifican la categoria de los caracteres `_` (U+005F) y `:` (U+003A) como “letras” (`\catcode`→P.20, 11) para que pueda escribir el ⟨*código*⟩ en lenguaje LaTeX3.

⟨*código*⟩ los espacios son ignorados, para ingresar un espacio debe usar `~`.

Loremipsum dolor

```
\ExplSyntaxOn
Lorem ipsum~dolor
\ExplSyntaxOff
```

Lorem ipsum

```
\def\macro{Lorem ipsum}

\ExplSyntaxOn
\use:c { m a cr o}
\ExplSyntaxOff
```

El ⟨*código*⟩ `\use:c { m a cr o}` es equivalente a componer ‘`\macro`’ usando `\csname`→P.214 y `\endcsname`.

6.2. Estructura léxica

6.2.1. Tipología de macro – Función y variable

El lenguaje clasifica una macro como una función o una variable.

\⟨*module*⟩_⟨*fun-name*⟩:⟨*arg-type*⟩

Identificador de una función [27, p. 3]. Por ejemplo, `\cs_new:Npn`.

\⟨*scope*⟩_⟨*module*⟩_⟨*var-name*⟩_⟨*type*⟩

Identificador de una variable. Por ejemplo, `\c_sys_jobname_str`.

6.2.2. Variable – Modificadores de acceso

Variables cuyo valor: [27, p. 5] [25, p. 2]

Capítulo 7

Composición extendida

Composición tipográfica extendida
—Typesetting extends, i.e., packages

La composición extendida cubre las capacidades fuera del LaTeX estandar. Al igual que la instrucción `import` o `using` para los lenguajes Java y C# respectivamente, el uso de una dependencia en LaTeX se declara mediante `\usepackage`→P.12.

7.1. Paquete `lipsum`

`\usepackage[⟨opción⟩]{lipsum}`

Este paquete v2.3 (2021-03-03, lipsum.pdf→P.710, .sty) provee 150 párrafos del texto mudo *Lorem Ipsum*.

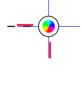
Dependencias: `expl3`→P.261 (2018/10/31) y `xparse`→P.288.

El paquete acepta la ⟨*opción*⟩ `nopar`, en este caso los párrafos se unen en un solo párrafo; entonces los comandos `\lipsum` y `\lipsum*` funcionan de manera opuesta.

La implementación (lipsum.ltd.tex→P.710, Ln: 30) del primer párrafo es:

```
\NewLipsumPar{%
  Lorem ipsum dolor sit amet, consectetuer
  adipiscing elit. Ut purus elit, vestibulum ut, placerat ac,
  adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu
  libero, nonummy eget, consectetuer id, vulputate a, magna. Donec
  vehicula augue eu neque. Pellentesque habitant morbi tristique
  senectus et netus et malesuada fames ac turpis egestas. Mauris ut
  leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna
  fringilla ultrices.  Phasellus eu tellus sit amet tortor gravida
  placerat. Integer sapien est, iaculis in, pretium quis, viverra ac,
  nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus.
  Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla.
  Curabitur auctor semper nulla. Donec varius orci eget risus. Duis
  nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis
  eget orci sit amet orci dignissim rutrum.}
```

el cual tiene 18 oraciones, y el segundo párrafo con 12 oraciones.



\lipsum*[⟨*paragraph*⟩][⟨*sentence*⟩]

Donde ⟨*sentence*⟩ es el número o rango de oración. ⟨*paragraph*⟩ es el número o rango de párrafo (lipsum.pdf→P.710, p. 3). El comando sin argumentos (por default), `\lipsum` imprime, los párrafos 1 al 7.

⟨*sentence range*⟩: expandir el párrafo 1, oración 2 al 4.

Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

```
\lipsum[1][2-4]
```

⟨*paragraph range*⟩: expandir los párrafos como párrafos múltiples.

Etiam vel ipsum. Morbi facilisis vestibulum nisl. Praesent cursus laoreet felis. Integer adipiscing pretium orci. Nulla facilisi. Quisque posuere bibendum purus. Nulla quam mauris, cursus eget, convallis ac, molestie non, enim. Aliquam congue. Quisque sagittis nonummy sapien. Proin molestie sem vitae urna. Maecenas lorem. Vivamus viverra consequat enim.
Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

```
\lipsum[65-66]
```

⟨*paragraph range*⟩ y la forma '*'. `\lipsum*` imprime los párrafos como un solo párrafo, i.e., hace unión (∪) de párrafos.

Etiam vel ipsum. Morbi facilisis vestibulum nisl. Praesent cursus laoreet felis. Integer adipiscing pretium orci. Nulla facilisi. Quisque posuere bibendum purus. Nulla quam mauris, cursus eget, convallis ac, molestie non, enim. Aliquam congue. Quisque sagittis nonummy sapien. Proin molestie sem vitae urna. Maecenas lorem. Vivamus viverra consequat enim. Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

```
\lipsum*[65-66]
```

⟨*paragraph range*⟩ y ⟨*sentence range*⟩. Párrafo 1 ∪ 2, oración 28 al 30.

Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

```
\lipsum[1-2][28-30]
```

7.2. Paquete `amsmath`

\usepackage[⟨*opciones*⟩]{amsmath}

Este paquete, v2.17a (2017/09/02, amsmath.pdf→P.696, .sty) desarrollado por la AMS, proporciona caracteristicas adicionales para facilitar la composición matemática. Un número de pequeños paquetes auxiliares también son distribuídos con el paquete `amsmath`.

Dependencias: `amstext`[1995/01/25], `amsbsy`[1995/01/20] y `amsopn`[1995/01/20].

Capítulo 8

Composición extendida (Working Draft)

8.1. Paquete amsfonts

\usepackage{amsfonts}

Este paquete v3.01 (2013/01/14, amsfonts.pdf→P.695, .sty)

Al igual que el paquete latexsym→P.366, este implementa los símbolos remobidos por la versión base §3.5.7→P.165.

\mho, \Join, \Box, \Diamond, \leadsto, \sqsubset, \sqsupset, \lhd, \unlhd, \rhd, \unrhd.

\mathbb{⟨*token*⟩}

Este comando (amsfonts.sty→P.689, ln: 108) es similar a \mathrm→P.151 pero produce Blackboard bold.

$\mathbb{N}$ —Natural numbers
$\mathbb{Z}$ —Integers
$\mathbb{Q}$ —Rational numbers
$\mathbb{R}$ —Real numbers
$\mathbb{C}$ —Complex numbers
$\mathbb{A}$ —Affine space
$\mathbb{P}$ —Projective space

```
$\mathbb{N}$ ---Natural numbers    \\
$\mathbb{Z}$ ---Integers           \\
$\mathbb{Q}$ ---Rational numbers   \\
$\mathbb{R}$ ---Real numbers       \\
$\mathbb{C}$ ---Complex numbers    \\
$\mathbb{A}$ ---Affine space       \\
$\mathbb{P}$ ---Projective space
```

Atributos de fuente *symbol font*§3.4.1→P.150 creados con \DeclareSymbolFont→P.243 en (amsfonts.sty→P.689, ln: 59)

- AMSa —Use font: {U}{msa}{m}{n};

=∠ =∠

```
={\usefont{U}{msa}{m}{n}\char92}
=$\angle$
```

- AMSb —Use font: {U}{msb}{m}{n};

℧ =℧

```
{\usefont{U}{msb}{m}{n}\char102}
=$\mho$
```

8.2. Paquete latexsym

\usepackage{latexsym}

Este paquete v2.2e (1998/08/17, latexsym.sty→P.690, .sty) provee las siguientes funciones:

- lasy —Use font: {U}{lasy}{m/b}{n};
  Atributos de fuente *symbol font*§3.4.1→P.150 creados con \DeclareSymbolFont→P.243 y \SetSymbolFont→P.243:

℧
m=℧
b=℧

```
$\mho$\\
m={\usefont{U}{lasy}{m}{n}\char48}\\
b={\usefont{U}{lasy}{b}{n}\char48}
```

- Al igual que el paquete amsfonts→P.365, implementa (con \DeclareMathSymbol→P.243) los símbolos remobidos por la versión base §3.5.7→P.165.

Tabla 8.1. Símbolos matemáticos paquete latexsym

Símbolo	Comando	Unicode	Type	Symbol font	Slot
℧	\mho		\mathord	lasy	48
⋈	\Join		\mathrel	lasy	49
□	\Box		\mathord	lasy	50
◇	\Diamond		\mathord	lasy	51
⇝	\leadsto		\mathrel	lasy	59
⊏	\sqsubset		\mathrel	lasy	60
⊐	\sqsupset		\mathrel	lasy	61
⊲	\lhd		\mathbin	lasy	1
⊴	\unlhd		\mathbin	lasy	2
⊳	\rhd		\mathbin	lasy	3
⊵	\unrhd		\mathbin	lasy	4

Capítulo 9

Composición gráfica con TikZ & PGF

Composición gráfica vectorial extendida
—Extended vectorial graphic composition

9.1. Introducción

9.1.1. Paquete TikZ

\usepackage{tikz}

Este paquete v3.0.1a (2015/08/07, tikz.sty→P.707, pgfmanual.pdf) [34, p. 128] provee un lenguaje declarativo y expresivo de alto nivel para crear gráficos vectoriales de forma nativa para LaTeX.

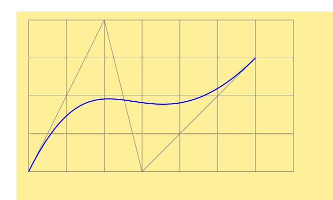
Dependencias del paquete: pgf→P.447 y pgffor→P.513.

Note que esta implícito que este paquete es necesario para todos los dibujos de este capítulo. También se da por implícito que este paquete indirectamente requiere el paquete xcolor→P.329.

9.1.2. Modo dibujo

\begin{tikzpicture}[⟨*opciones*⟩]
⟨*contenido del entorno*⟩
\end{tikzpicture}

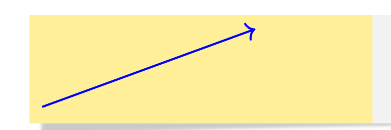
Este entorno [34, p. 128] se usa para dibujar un gráfico vectorial. En el ⟨*contenido del entorno*⟩ se activa el modo dibujo, en el, se escriben las instruciones (una serie de secuencia de tokens) que sirven para producir el gráfico. Las ⟨*opciones*⟩ son atributos de dibujo §9.5→P.421.



```
\begin{tikzpicture}[scale=.5]
  \draw[help lines]    (0,0)         grid          (7,4);
  \draw[help lines]    (0,0) -- (2,4)  -- (3,0) -- (6,3);
  \draw[blue] (0,0) .. controls (2,4) and (3,0) .. (6,3);
\end{tikzpicture}
```

\tikz[⟨*opciones*⟩]⟨*instruciones de dibujo*⟩

Este comando [34, p. 130] es similar al entorno tikzpicture pero sirve donde las ⟨*instruciones de dibujo*⟩ solo requieren una o dos líneas de código.



```
\tikz \path[->,thick,draw=blue] (0,0) -- (20:3);
```

9.1.3. Comandos de dibujo

`\path`⟨*especificación*⟩;

Este comando [34, p. 153] dibuja una serie de gráficos vectoriales según la ⟨*especificación*⟩ de funciones de dibujo §9.4→P.401.

`\draw`

es una abreviación para `\path[draw]`, véase atributo `draw`→P.422.

`\fill`

es una abreviación para `\path[fill]`, véase atributo `fill`→P.424.

`\filldraw`

es una abreviación para `\path[fill,draw]`.

`\pattern`

es una abreviación para `\path[pattern]`, véase atributo `pattern`→P.427.

`\shade`

es una abreviación para `\path[shade]`, véase atributo `shade`→P.428.

`\shadedraw`

es una abreviación para `\path[shade,draw]`.

`\clip`

es una abreviación para `\path[clip]`.

`\useasboundingbox`

es una abreviación para `\path[use as bounding box]`.

`\node`

Este comando [34, p. 227] es una abreviación para `\path node`, véase función de dibujo `node`→P.413.

`\coordinate`

Este comando [34, p. 227] es una abreviación para `\path coordinate`, véase función `coordinate`→P.416.

`\pic`

Este comando [34, p. 262] es una abreviación para `\path pic`, véase función `pic`→P.417.

`\graph`

Este comando [34, p. 273] es una abreviación para `\path graph`, véase función `graph`→P.420.

9.1.4. Atributo de dibujo

`\begin{scope}[`⟨*opciones*⟩`]`
⟨*contenido del entorno*⟩
`\end{scope}`

Este entorno [34, p. 131] hace que los atributos de dibujo, ⟨*opciones*⟩, sean visibles dentro del ⟨*contenido del entorno*⟩.

```
\begin{tikzpicture}[ultra thick]
    \begin{scope}[red]
        \draw (0,0) -- (1,1);
        \draw (1,0) -- (2,1);
    \end{scope}
    \draw (3,0) -- (4,1);
    \begin{scope}[teal]
        \draw (5,0) -- (6,1);
        \draw (6,0) -- (7,1);
        \draw[blue] (7,0) -- (8,1);
    \end{scope}
\end{tikzpicture}
```

`\tikzset{⟨opciones⟩}`

Este comando [34, p. 133] permite gestionar ⟨*opciones*⟩ o atributos §9.5.29→P.445 usando el comando `\pgfkeys`. Por ejemplo, usando el gestor de atributo `.style`→P.446 se define el atributo `vector`:

```
\tikzset{vector/.style={draw=#1,very thick,->}}
\begin{tikzpicture}
    \draw [vector=red]  (0,0) -- (1,1);
    \draw [vector=blue] (1,0) -- (2,1);
\end{tikzpicture}
```

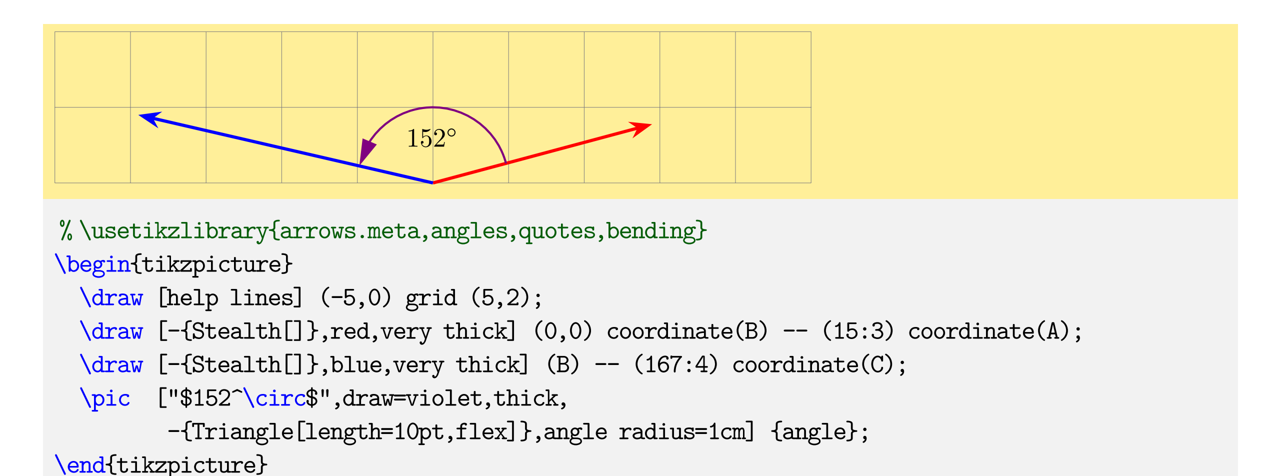
Visibilidad: El atributo definido es visible solo en el grupo local de LATEX.

```
\tikzset{ivec/.style={draw=red,very thick,->}}
{
    \tikzset{jvec/.style={draw=blue,very thick,<-}}
    \tikz\draw [ivec]  (0,0) -- (1,1);
    \tikz\draw [jvec]  (0,0) -- (1,1);
}
\tikz\draw [ivec]  (0,0) -- (1,1);
%\tikz\draw [jvec]  (0,0) -- (1,1); %% No compila, no visible
```

9.1.5. Librerias

`\usetikzlibrary{⟨lista de librerias⟩}`

Este comando carga la ⟨*lista de librerias*⟩ §9.8→P.471 del paquete. Los nombres de cada libreria debe estar separdada por comas.




```
% \usetikzlibrary{arrows.meta,angles,quotes,bending}
\begin{tikzpicture}
  \draw [help lines] (-5,0) grid (5,2);
  \draw [-{Stealth[]},red,very thick] (0,0) coordinate(B) -- (15:3) coordinate(A);
  \draw [-{Stealth[]},blue,very thick] (B) -- (167:4) coordinate(C);
  \pic  ["$152^\circ$",draw=violet,thick,
         -{Triangle[length=10pt,flex]},angle radius=1cm] {angle};
\end{tikzpicture}
```

Para el ejemplo, las librerias utilizadas son:

- Atributo `-{Stealth[]}` y `-{Triangle[]}`; responsable, libreria `arrows.meta`→P.473.
- Atributo `flex`; responsable, libreria `bending`→P.475.
- Atributo `"$152^\circ$"`; responsable, libreria `quotes`→P.510.
- Atributo `angle radius`; responsable, libreria `angles`→P.511.

9.2. Cómputo matemático

Procesamiento de expresiones, operadores, y funciones matemáticas.

9.2.1. Expresión matemática

`\pgfmathparse`{⟨*expresión*⟩}

Esta macro [34, p. 1028] analiza la ⟨*expresión*⟩ y retorna el resultado como un escalar por la macro `\pgfmathresult`.

152.0.

```
\pgfmathparse{167-15} \pgfmathresult.
```

La operación entre distintas unidades de longitud, se convierte a puntos TeX (`pt`) [34, p. 1030] y retorna el valor sin unidades.

799.35971.

```
\pgfmathparse{177mm+2*0.3333em+4in} \pgfmathresult.
```

9.2.2. Constantes

`pi` = 3.141592654

`e` = 2.718281828

`true` = 1

Capítulo 10

Proyecto MDTZiØFEX

Contribución, experiencia e investigación
—Advancing research. Administrative experience

Este proyecto es el cimiento de la presente obra. Provee interfaces para su escritura (a nivel de código), regula su estructura (a nivel renderizado), monitorea su evolución (estadísticas de progreso), y realiza investigación avanzada (lógica, matemática y programación) para enriquecer el contenido del documento principal. Técnicamente la obra le debe su existencia a este proyecto.

10.1. Introducción

El proyecto MDTZiØFEX se crea para dar soporte a las nuevas obras literarias de MDT496 ESTUDIOS.

10.1.1. Logos

El logo se define por: **MDT** (MDT496 ESTUDIOS); **ZIO** (mathematical); **FEX** (programming).

```
% MDTZiOFex
%    760                      496 octal
% 111110000                   496 binario
%     01F0                    496 hexadecimal
%        ex(te|pa)nds         Conciso y expresivo
```

`\mdtziofex`

Este comando robusto (mdtziofex-main.tex→P.585, Ln: 12) expande el logo del proyecto MDTZiØFEX.

```
\protected\gdef\mdtziofex{\leavevmode\hbox{\m@th
    \if b\expandafter\@car\f@series\@nil\boldmath\fi
    MDT\kern-.26em%
    {\usefont{OML}{cmm}{b}{it}\char`7}%
    \lower.19444em\hbox to 0pt{%
          \hss{\vrule width.7em height.05em}\hss}%
    $\mathchar379$%
    \lower.19444em\hbox{\usefont{OMS}{cmsy}{b}{n}\char59}%
    F\kern-.1667em{\usefont{OT1}{cmtt}{m}{sc}ex}}}
```

MDTziofex	`\mdtziofex`

basado en los logos de TeX, LaTeX y LaTeX 2_ε, es escalable en modo párrafo y matemático.

MDTziofex	`{\huge\bfseries\mdtziofex}`

10.1.2. Terminología Unicode

Para evitar ambigüedad de los caracteres que se escriben en el código fuente, se emplea como referencia el estándar Unicode [10] para identificar un carácter visible (e.g., β (U+03B2)) y un carácter de control (e.g., CR (U+000D)).
Por ejemplo, cuando copia el siguiente código fuente

Lorem' ipsum Lorem' ipsum	`Lorem' ipsum\par` `Lorem’ ipsum`

debe notar[1] que en el código fuente se teclea los caracteres ' (U+0027) y ’ (U+2019). Las propiedades Unicode [10, p. 869] de estos se muestran en la Tabla 10.1.

Tabla 10.1. Análisis de codificación de caracteres Unicode

Glifo	Unicode	Name	Type	Block
'	0027	APOSTROPHE	Other Punctuation (Po)	Basic Latin (0000)
’	2019	RIGHT SINGLE QUOTATION MARK	Final Quote Punctuation (Pf)	General Punctuation (2000)

10.2. Libreria para documentar macro

MDTziofex —*Macro Documentation Support*

Proyecto MDTziofex, módulo *Macro Documentation Support* (`texspecs`→P.529). Este módulo provee interfaces (macros, entornos, etc.) para escribir la documentación de una macro. Estas son implementadas cuidadosamente, con soporte hasta el nivel de lenguaje LaTeX3.

10.2.1. Abreviaturas

`\cmd`⟨*control sequence*⟩

Command ⟨*control sequence*⟩. `\cmd` es un comando robusto.

prints `\macro`.	`prints \cmd\macro.`

Soporta comandos LaTeX3

prints `\cs_new:Npn`.	`prints \cmd{\cs_new:Npn}.`

En algunos casos extremos[2] puede recurrir a / en lugar de \.

[1] El compilado produce el mismo glifo, esto es tema de codificación del archivo (UTF-8, véase `inputenc`→P.343) y fuente (OT1, véase `fontenc`→P.349) que no se trata aquí.
[2] por ejemplo el comando `\par`→P.25 no puede escribirse directamete

prints \par.	`prints \cmd{/par}.`

`\stripcommand`⟨*control sequence*⟩

Strip command ⟨*control sequence*⟩. Desmantelar comando.

prints macro	`prints \stripcommand\macro`

`\cs{`⟨*name*⟩`}`

Control sequence ⟨*name*⟩. `\cs` es robusto.

prints \macro.	`prints \cs{macro}.`

`\removeats{`⟨*control sequence*⟩`}`

Remueve el backslash y reemplaza los caracteres `@` y `_` por los comandos `\mcharAT` y `\mcharLOWLINE` respectivamente.

print "___cs___@n@ew___error"
print "XXcsXXYnYewXXerror"

```
\removeats{\__cs__@n@ew__error}
print ``\strippedat''

{
    \def\mcharLOWLINE{X}
    \def\mcharAT{Y}
    print ``\strippedat''
}
```

10.2.2. Argumentos

`\meta{`⟨*text*⟩`}`

Argument ⟨*text*⟩ token.

prints ⟨*text*⟩.	`prints \meta{text}.`
angle=⟨*A*⟩-⟨*B*⟩-⟨*C*⟩	`{\ttfamily angle=\meta{A}--\meta{B}--\meta{C}}`
$\langle U\rangle^4 = \sin\langle\theta\rangle$	`$\meta{U}^4=\sin \meta{$\theta$}$`

`\marg{`⟨*text*⟩`}`

Mandatory argument.

prints {⟨*text*⟩}.	`prints \marg{text}.`

`\oarg{`⟨*text*⟩`}`

Optional argument.

10.8. Investigación

10.8.1. The LaTeX Language Specification (Synthesis)

Este manual tiene como objetivo general de especificar los comandos del lenguaje LaTeX; por herencia se especifica los comandos de TeX Primitive; finalmente, por correlación (i.e., comandos en comun) se especifica algunos comandos del lenguaje Plain TeX.

Tabla 10.2. The LaTeX Language Specification (Synthesis)

Lenguaje	Progreso	comandos especificados
TeX primitive→P.588	237/321	73.0 %
Plain TeX→P.592	496/518	95.0 %
LaTeX 2$_\varepsilon$→P.597	905/974	92.0 %
LaTeX3→P.607	57/190	30.0 %
—*Consolidado*—	1256/1550	81.0 %

La Tabla 10.2 detalla el progreso de avance de la cantidad de comandos especificados para un nivel de lenguaje. Las secuencias de control que implementan según el nivel del lenguaje se describen en la siguiente lista:

1. Síntesis lenguaje TeX primitive, Tabla 10.3 (321 comandos).

   Comandos primitivos del lenguaje TeX, donde Knuth [1, p. 10] indica son 300 comandos, pero alguno de ellos fueron redefinidos por los diferentes lenguajes que heredan de el.

2. Síntesis lenguaje Plain TeX, Tabla 10.4 (518 comandos).

   'Macros Básicos' originales [7, p. 14] implementadas por Knuth [1, p. 10], que indica que son cerca de 600 comandos, y descritas en su Apéndice B [1, p. 339].

3. Síntesis lenguaje LaTeX 2$_\varepsilon$, Tabla 10.5 (974 comandos).

   Las macros, comandos (robustos) y entornos primitivos del lenguaje LaTeX 2$_\varepsilon$ [7].

   La tabla contiene todos los comandos si discriminación (LaTeX y LaTeX 2$_\varepsilon$), sin embargo LaTeX 2$_\varepsilon$ es una actualización [3] [2, p. 227], por tanto provee comandos nuevos y deja obsoleto a otros. Por ejemplo, provee el comando `\tt`→P.48 de TeX Plain y remueve a `\mho`→P.366 de LaTeX 2.09.

4. Síntesis lenguaje LaTeX3, Tabla 10.6 (190 comandos).

   La tabla, aun en desarrollo, contiene solo algunos de los identificadores (funciones y variables) del lenguaje LaTeX3.

Tabla 10.3. Síntesis lenguaje TeX primitive (321 comandos)

TeX primitive	TeX primitive	TeX primitive
`\above`→P.175	`\abovedisplayshortskip`→P.149	`\abovedisplayskip`→P.149
`\abovewithdelims`→P.175	`\accent`→P.214	`\adjdemerits`→P.??
`\advance`→P.201	`\afterassignment`→P.213	`\aftergroup`→P.214

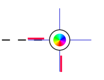


Tabla 10.3. (Continua de la página anterior)

TEX primitive	TEX primitive	TEX primitive
\atop→P.175	\atopwithdelims→P.175	\badness→P.??
\baselineskip→P.32	\batchmode→P.254	\begingroup→P.23
\belowdisplayshortskip→P.149	\belowdisplayskip→P.149	\binoppenalty→P.41
\botmark→P.135	\box→P.209	\boxmaxdepth→P.??
\brokenpenalty→P.41	\catcode→P.20	\char→P.19
\chardef→P.200	\cleaders→P.38	\closein→P.??
\closeout→P.129	\clubpenalty→P.41	\copy→P.210
\count→P.199	\countdef→P.200	\cr→P.77
\crcr→P.77	\csname→P.214	\day→P.129
\deadcycles→P.??	\def→P.187	\defaulthyphenchar→P.43
\defaultskewchar→P.??	\delcode→P.??	\delimiter→P.??
\delimiterfactor→P.??	\delimitershortfall→P.??	\dimen→P.199
\dimendef→P.200	\discretionary→P.42	\displayindent→P.148
\displaylimits→P.??	\displaystyle→P.153	\displaywidowpenalty→P.41
\displaywidth→P.148	\divide→P.202	\doublehyphendemerits→P.??
\dp→P.194	\edef→P.190	\else→P.204
\emergencystretch→P.??	\end→P.15	\endcsname→P.214
\endgroup→P.23	\endinput→P.127	\endlinechar→P.??
\eqno→P.147	\errhelp→P.251	\errmessage→P.251
\errorcontextlines→P.258	\errorstopmode→P.254	\escapechar→P.21
\everycr→P.77	\everydisplay→P.145	\everyhbox→P.194
\everyjob→P.125	\everymath→P.145	\everypar→P.25
\everyvbox→P.194	\exhyphenpenalty→P.41	\expandafter→P.211
\fam→P.215	\fi→P.204	\finalhypendemerits→P.??
\firstmark→P.135	\floatingpenalty→P.41	\font→P.215
\fontdimen→P.216	\fontname→P.215	\futurelet→P.??
\gdef→P.190	\global→P.211	\globaldefs→P.??
\halign→P.78	\hangafter→P.??	\hangindent→P.??
\hbadness→P.??	\hbox→P.191	\hfil→P.36
\hfill→P.36	\hfilneg→P.37	\hfuzz→P.??
\hoffset→P.??	\holdinginserts→P.??	\hrule→P.92
\hsize→P.210	\hskip→P.35	\hss→P.37
\ht→P.193	\hyphenation→P.43	\hyphenchar→P.43
\hyphenpenalty→P.41	\if→P.204	\ifcase→P.208

Print Preview (Beta Test) ~ Presentación Preliminar